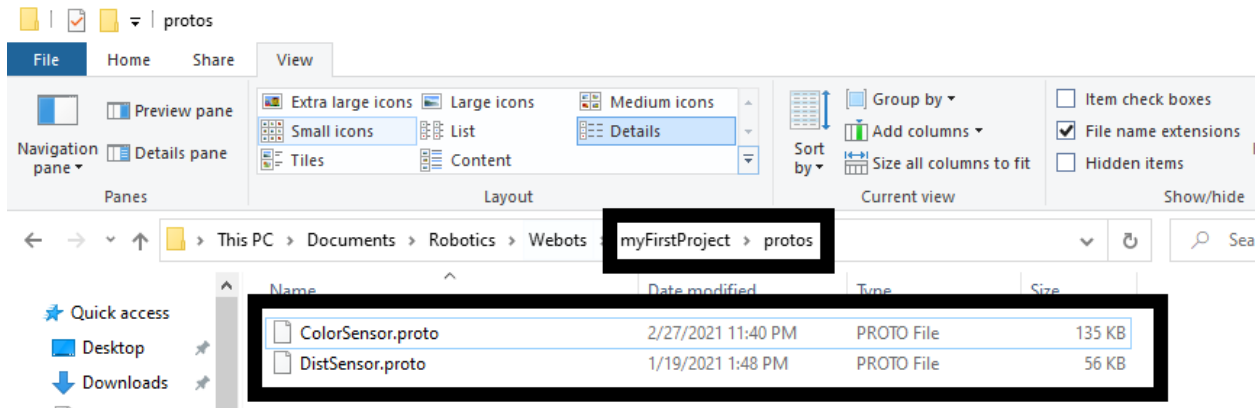


To add better distance + color sensor:

When you go to add a sensor, instead of using “Based Nodes”, use “PROTO nodes (Current Project)” to add a distance or color sensor. You will actually be able to see the sensors using this method (the base sensors in Webots are all invisible) and they will have friendly values. The distance sensor will read in centimeters from 1 cm to 1000 cm.

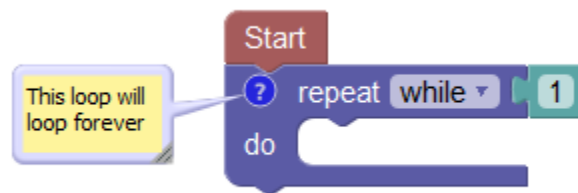


Note: If you are using a world in the SR-Blockly folder (which you should be doing). There will be multiple sensors on the robot already. Go ahead and delete those sensors and add your own sensors.

Distance Sensor Challenges:

Challenge 1:

Print out the distance sensor value in a forever loop (hint: if you use a while loop and attach the number 1 to it, it will loop forever).



Challenge 2:

Have the robot move forward and stop when it reaches a certain distance from the wall (you decide the distance).

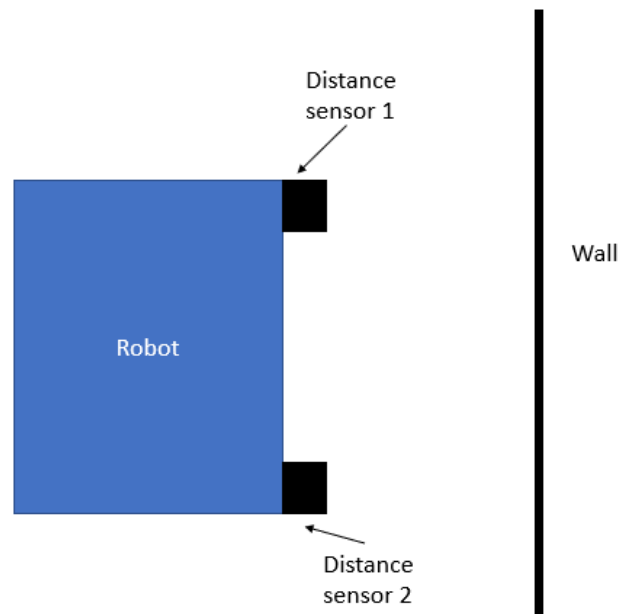
Challenge 3:

Repeat challenge 2 but have the robot gradually slow down as it approaches the target distance (hint: if you do this in a clever way you shouldn't need many lines of code).

Challenge 4:

IMPORTANT: Before you program this challenge, you MUST write a flowchart for this challenge and get it checked by Aditya or Jeff. You can use draw.io, but if you really want to use another software that is fine.

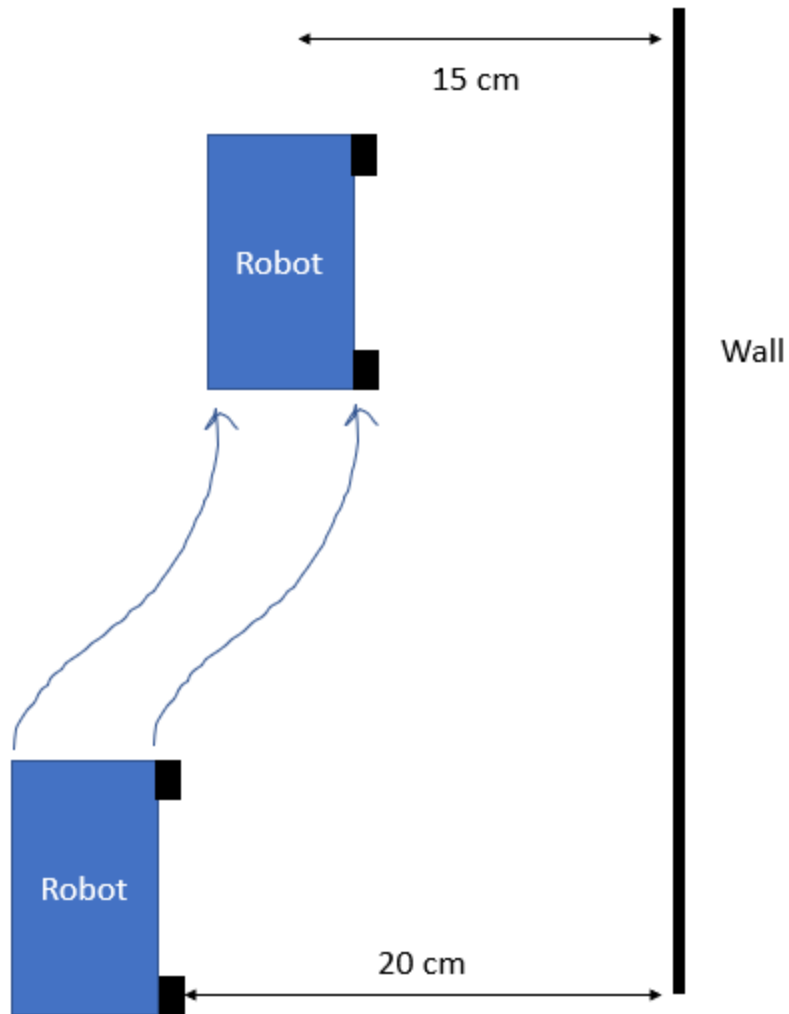
Add two sensors to the right side of the robot – one in the front and one in the back. Write a program that will always keep the robot parallel to the wall – rotate the robot and it should adjust itself back.



A figure for the set-up for challenge 4 – the positions don't have to match exactly, but it shows the general idea

Challenge 5: (tricky)

Repeat challenge 4 but try to have the robot maintain a certain distance from the wall. For example, if you are trying to maintain 15 cm from the wall but your robot is 20 cm away, your robot should tilt itself closer to the wall then realign itself with the wall when it gets close to 15 cm (see picture on next page). This should be a continuous process, so your robot should continue to go forward and adjust as necessary – it will essentially be “wall tracing” (following the right-hand wall).



Color sensor challenges:

Challenge 6:

Add a color sensor onto your robot and continually print out the gray value of the color sensor (this is basically like an NXT light sensor).

Challenge 7:

If you reach this point, privately message Jeffrey on Slack because there is a file you need before you keep going.

IMPORTANT: Before you program this challenge, you **MUST** write a flowchart for this challenge and get it checked by Aditya or Jeff. You can use draw.io, but if you really want to use another software that is fine.

Open up the world line1.wbt and write a line tracing program using one color sensor.

Challenge 8:

IMPORTANT: Before you program this challenge, you MUST write a flowchart for this challenge and get it checked by Aditya or Jeff. You can use draw.io, but if you really want to use another software that is fine.

Open up the world obstacle.wbt. The goal is to line trace around the track (using one color sensor) while avoiding the obstacle (using one distance sensor).